

Package ‘err’

October 13, 2022

Title Customizable Object Sensitive Messages

Version 0.2.0

Description Messages should provide users with readable information about R objects without flooding their console.
'cc()' concatenates vector and data frame values into a grammatically correct string using commas, an ellipsis and conjunction.
'cn()' allows the user to define a string which varies based on a count.
'co()' combines the two to produce a customizable object aware string.
The package further facilitates this process by providing five 'sprintf'-like types such as '%n' for the length of an object and '%o' for its name as well as wrappers for pasting objects and issuing errors, warnings and messages.

License MIT + file LICENSE

Suggests testthat, covr

URL <https://github.com/poissonconsulting/err>

BugReports <https://github.com/poissonconsulting/err/issues>

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Author Joe Thorley [aut, cre] (<<https://orcid.org/0000-0002-7683-4592>>),
James Dunham [aut]

Maintainer Joe Thorley <joe@poissonconsulting.ca>

Repository CRAN

Date/Publication 2019-04-25 20:50:20 UTC

R topics documented:

cc	2
cn	3
co	4

err	5
err_wrappers	6
msg	7
p	7
wrn	8

Index	9
--------------	----------

cc	<i>Concatenation with Commas</i>
----	----------------------------------

Description

Concatenates object values into a string with each value separated by a comma and possibly the last value separated by a conjunction.

Usage

```
## Default S3 method:
cc(object, conjunction = NULL, bracket = "",
    ellipsis = 10, oxford = FALSE, ...)

## S3 method for class 'character'
cc(object, conjunction = NULL, bracket = "'",
    ellipsis = 10, oxford = FALSE, ...)

## S3 method for class 'factor'
cc(object, conjunction = NULL, bracket = "'",
    ellipsis = 10, oxford = FALSE, ...)

## S3 method for class 'data.frame'
cc(object, conjunction = NULL, ellipsis = 10,
    oxford = FALSE, ...)
```

Arguments

object	The object with values to concatenate.
conjunction	A string of the conjunction to separate the last value by or NULL.
bracket	A string to bracket the values by.
ellipsis	A count of the total number of values required to use an ellipsis.
oxford	A flag indicating whether to use the Oxford comma (if conjunction).
...	Unused

See Also

[co](#)

Examples

```

cc(c(1,1,1:2))
cc(100:1)
cc(1:100, "and")
cc(100:1, "or", bracket = "|", ellipsis = 5, oxford = TRUE)
cc(mtcars)

```

cn

Customizable Number Aware String

Description

Customizable Number Aware String

Usage

```

cn(n, one = "there %r %n value%s", some = one, none = some,
   lots = some, nlots = 10)

```

Arguments

n	A count of the number.
one	The string to return if n = 1
some	The string to return if n is in 2, 3, ..., nlots - 1
none	The string to return if n = 0
lots	The string to return if n >= nlots
nlots	A count of the number of values to consider to be lots

Value

A string of the updated message.

sprintf-like types

The following sprintf-like types can be used in the custom messages:

```

n the length of the object
s 's' if n != 1 otherwise ''

```

See Also

[cc](#) and [co](#)

Examples

```

cn(0)
cn(1)
cn(4)

```

Description

Produces a fully customizable object aware string with consecutive values separated by columns.

Usage

```
## Default S3 method:
co(object, one = "%o has %n value%s: %c",
    some = one, none = gsub(":", "", some), lots = some, nlots = 10,
    conjunction = NULL, bracket = "", ellipsis = nlots,
    oxford = FALSE, object_name = substitute(object), ...)

## S3 method for class 'character'
co(object, one = "%o has %n value%s: %c",
    some = one, none = gsub(":", "", some), lots = some, nlots = 10,
    conjunction = NULL, bracket = "", ellipsis = nlots,
    oxford = FALSE, object_name = substitute(object), ...)

## S3 method for class 'factor'
co(object, one = "%o has %n value%s: %c",
    some = one, none = gsub(":", "", some), lots = some, nlots = 10,
    conjunction = NULL, bracket = "", ellipsis = nlots,
    oxford = FALSE, object_name = substitute(object), ...)

## S3 method for class 'data.frame'
co(object, one = "%o has %n column%s\n%c",
    some = one, none = none, lots = some, nlots = 10,
    conjunction = NULL, ellipsis = nlots, oxford = FALSE,
    object_name = substitute(object), ...)
```

Arguments

object	The object of length n
one	The string to return if n = 1
some	The string to return if n is in 2, 3, ..., nlots - 1
none	The string to return if n = 0
lots	The string to return if n >= nlots
nlots	A count of the number of values to consider to be lots
conjunction	A string of the conjunction to separate the last value by or NULL.
bracket	A string to bracket the values by.
ellipsis	A count of the total number of values required to use an ellipsis.

oxford	A flag indicating whether to use the Oxford comma (if conjunction).
object_name	A string of the object name.
...	Unused.

sprintf-like types

The following sprintf-like types can be used in the custom messages:

- c the object as a comma separated list (produced by a [cc](#) function)
- n the length of the object
- o the name of the object
- s 's' if n != 1 otherwise ''
- r 'are' if n != 1 otherwise 'is'

See Also

[cc](#)

Examples

```
co(character())
x <- "fox"
co(x)
co(c(1,2,5))
co(1:10)
co(datasets::mtcars)
```

err

Error

Description

Stops execution and throws an error without the call as part of the error message.

Usage

```
err(...)
```

Arguments

... zero or more objects which can be coerced to character

See Also

[stop](#), [wrn](#), [msg](#) and [co](#)

err_wrappers

Wrappers with Conjunction and " Brackets.

Description

Wrappers with Conjunction and " Brackets.

Usage

```
cc_and(object)
```

```
cc_or(object)
```

```
co_and(object, one = "%o has %n value%s: %c",  
        object_name = substitute(object))
```

```
co_or(object, one = "%o has %n value%s: %c",  
        object_name = substitute(object))
```

Arguments

object	The object with values to concatenate.
one	The string to return if n = 1
object_name	A string of the object name.

Functions

- `cc_and`: Concatenation with Commas and And Conjunction
- `cc_or`: Concatenation with Commas and Or Conjunction
- `co_and`: Customizable Object Aware String and And Conjunction
- `co_or`: Customizable Object Aware String and Or Conjunction

See Also

[cc](#) and [co](#)

Examples

```
cc_and(1:100)  
cc_or(100:1)  
co_and(1:100)  
co_or(100:1)
```

msg	<i>Message</i>
-----	----------------

Description

Generates a diagnostic message.

Usage

```
msg(...)
```

Arguments

... zero or more objects which can be coerced to character

See Also

[message](#), [err](#), [wrn](#) and [and](#) [co](#)

p	<i>Paste</i>
---	--------------

Description

Wrappers on [paste](#) and [paste0](#) to increase the readability of code.

Usage

```
p(..., sep = " ", collapse = NULL)
```

```
p0(..., collapse = NULL)
```

Arguments

... one or more R objects, to be converted to character vectors.
sep a character string to separate the terms. Not [NA_character_](#).
collapse an optional character string to separate the results. Not [NA_character_](#).

Value

A string of the pasted values.

Functions

- `p0`: `Paste0`

Examples

```
p("The", "red")  
p0("ard", "vark")
```

wrn

Warning

Description

Immediately outputs an warning without the call as part of the error message that as far as possible is a single line.

Usage

```
wrn(...)
```

Arguments

... zero or more objects which can be coerced to character

See Also

[warning](#), [err](#), [msg](#) and [co](#)

Index

cc, [2](#), [3](#), [5](#), [6](#)

cc_and(err_wrappers), [6](#)

cc_or(err_wrappers), [6](#)

cn, [3](#)

co, [2](#), [3](#), [4](#), [5–8](#)

co_and(err_wrappers), [6](#)

co_or(err_wrappers), [6](#)

err, [5](#), [7](#), [8](#)

err_wrappers, [6](#)

message, [7](#)

msg, [5](#), [7](#), [8](#)

NA_character_, [7](#)

p, [7](#)

p0 (p), [7](#)

paste, [7](#)

paste0, [7](#)

stop, [5](#)

warning, [8](#)

wrn, [5](#), [7](#), [8](#)